



Examen Final

*Aucun document n'est autorisé.
Tout appareil électronique doit être éteint (Téléphone, Ordinateur, Tablette, etc.).*

1 Gestion d'un CyberCafé

(14 pts, 55 min)

Un propriétaire d'un CyberCafé fait appel à vos compétences en programmation pour lui faire un programme permettant de facturer ses clients à la minute.

Une session internet est caractérisée par le numéro de poste, l'heure de début et l'heure de fin. L'exemple ci-contre montre la représentation d'une session d'un client qui s'est connecté au poste numéro 07 de 11 :05 :37 à 12 :27 :54.

poste	heure_deb	heure_fin
07	11:05:37	12:27:54

1. Créer un type de donnée permettant de déclarer des variables de type session. (3 pts)
2. Écrire une procédure permettant de lire une variable de type session. (3 pts)
3. Écrire une fonction qui permet de vérifier la validité d'une session. Cette fonction vérifie pour une session si l'heure de début est inférieur à l'heure de fin. (3 pts)
4. Écrire une fonction qui renvoie la durée d'une session sous forme de [heure :minute :seconde]. Par exemple, la durée de la session de la figure ci-dessus est [01 :22 :17]. (3 pts)
5. Écrire un programme principal qui demande à l'utilisateur d'introduire les données concernant une session puis calcule le tarif correspondant à la durée de cette session (1DA/min). (2 pts)

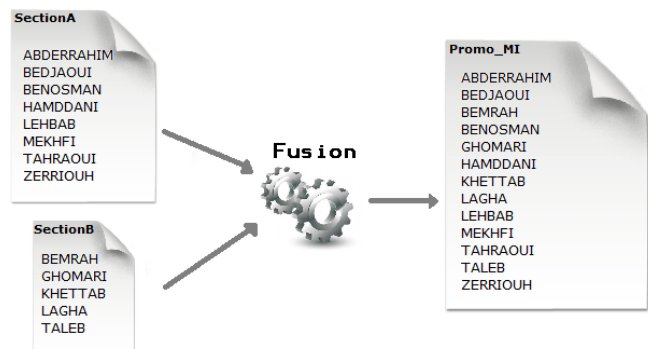
2 Fusion de fichiers

(6 pts, 35 min)

On dispose de deux fichiers textes contenant les noms des étudiants des deux sections MI (SectionA et SectionB). Chaque fichier contient exactement un seul nom par ligne (lettres majuscules sans espacement). Ces deux fichiers sont déjà triés dans l'ordre (lexicographique) croissant.

On se propose de fusionner ces deux fichiers en un fichier unique appelé Promo_MI, qui contiendra tous les noms qui se trouvent dans les des deux fichiers SectionA et SectionB et qui sera, lui aussi, trié dans l'ordre croissant.

Indice. On peut lire alternativement, parfois dans SectionA, parfois dans SectionB, et on décide à chaque fois quel nom on écrit dans le fichier Promo_MI.



1. Écrire la procédure qui permet de fusionner deux fichiers textes dans un troisième fichier avec le principe expliqué ci-dessus. Cette procédure prend trois chaînes de caractères en paramètres (les noms des trois fichiers) et s'occupe également de l'ouverture et de la fermeture des trois fichiers.
2. Écrire un programme principal pour tester cette procédure.

« Bon courage! »



Correction de l'Examen Final

*Aucun document n'est autorisé.
Tout appareil électronique doit être éteint (Téléphone, Ordinateur, Tablette, etc.).*

1 Gestion d'un CyberCafé

(14 pts, 55 min)

Un propriétaire d'un CyberCafé fait appel à vos compétences en programmation pour lui faire un programme permettant de facturer ses clients à la minute.

Une session internet est caractérisée par le numéro de poste, l'heure de début et l'heure de fin. L'exemple ci-contre montre la représentation d'une session d'un client qui s'est connecté au poste numéro 07 de 11 :05 :37 à 12 :27 :54.

poste	heure_deb	heure_fin
07	11:05:37	12:27:54

1. Créer un type de donnée permettant de déclarer des variables de type session. (3 pts)

Solution

```
1 PROGRAM CyberCafe;
2
3 TYPE
4   heure_t = 0..23;
5   minute_t = 0..59;
6   seconde_t = 0..59;
7   temps_t =
8     Record
9       h : heure_t;
10      m : minute_t;
11      s : seconde_t;
12     End;
13 session_t =
14   Record
15     poste : integer;
16     heure_deb : temps_t;
17     heure_fin : temps_t;
18   End;
19
20 VAR
21   c : session_t;
22   duree : temps_t;
```

2. Écrire une procédure permettant de lire une variable de type session. (3 pts)

Solution

```
1 Procedure Lire_temps (var t : temps_t);
2   Begin
3     write('heure_: '); readln(t.h);
4     write('minute_: '); readln(t.m);
5     write('seconde_: '); readln(t.s);
6   End;
```

```

1 Procedure Lire_session (var c : session_t);
2     Begin
3         write('Donner le numero de poste: ');
4         readln(c.poste);
5         writeln('Donner l''heure de debut de la session: ');
6         Lire_temps(c.heure_deb);
7         writeln('Donner l''heure de fin de la session: ');
8         Lire_temps(c.heure_fin);
9     End;

```

3. Écrire une fonction qui permet de vérifier la validité d'une session. Cette fonction vérifie pour une session si l'heure de début est inférieur à l'heure de fin. (3 pts)

Solution

```

1 Function Valide (c : session_t) : boolean ;
2     Var b : boolean ;
3     Begin
4         b := (c.heure_deb.h < c.heure_fin.h) or
5         (c.heure_deb.h = c.heure_fin.h) and
6         ((c.heure_deb.m < c.heure_fin.m) or (c.heure_deb.m = c.heure_fin.m) and
7         (c.heure_deb.s < c.heure_fin.s) );
8         Valide := b;
9     End;

```

4. Écrire une fonction qui renvoie la durée d'une session sous forme de [heure :minute :seconde]. Par exemple, la durée de la session de la figure ci-dessus est [01 :22 :17]. (3 pts)

Solution

```

1 Function Convertir_s ( t : temps_t) : LONGINT ;
2     Var s : LONGINT;
3     Begin
4         s := (t.h) * 3600 + (t.m) * 60 + t.s;
5         Convertir_s := s;
6     End;
7
8 Function Calcul_duree ( c : session_t) : temps_t ;
9     Var diff, ns_deb ,ns_fin : LONGINT ;
10    var duree : temps_t;
11    Begin
12    if (Valide(c)) then
13        Begin
14            ns_deb := Convertir_s (c.heure_deb);
15            ns_fin := Convertir_s (c.heure_fin);
16            diff := ns_fin - ns_deb;
17            duree.h := diff div 3600;
18            diff := diff mod 3600;
19            duree.m := diff div 60;
20            diff := diff mod 60;
21            duree.s := diff;
22        END
23    Else writeln('la session n''est pas valide');
24    Calcul_duree := duree;
25    End;

```

5. Écrire un programme principal qui demande à l'utilisateur d'introduire les données concernant une session puis calcule le tarif correspondant à la durée de cette session (1DA/min). (2 pts)

Solution

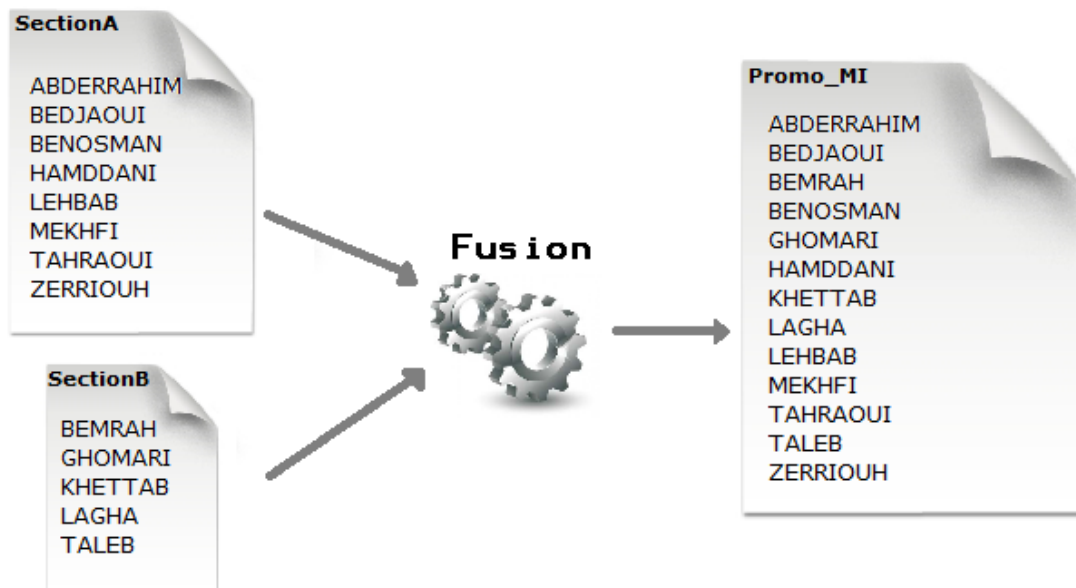
```
1 BEGIN
2 Lire_session(c);
3 duree := Calcul_duree(c);
4 writeln('la_duree_de_la_connexion_est:', duree.h, ':', duree.m, ':', duree.s );
5 Writeln('Tarif:', duree.h*60 + duree.m, 'DA');
6 END.
```

2 Fusion de fichiers

(6 pts, 35 min)

On dispose de deux fichiers textes contenant les noms des étudiants des deux sections MI (SectionA et SectionB). Chaque fichier contient exactement un seul nom par ligne (lettres majuscules sans espacement). Ces deux fichiers sont déjà triés dans l'ordre (lexicographique) croissant.

On se propose de fusionner ces deux fichiers en un fichier unique appelé Promo_MI, qui contiendra tous les noms qui se trouvent dans les des deux fichiers SectionA et SectionB et qui sera, lui aussi, trié dans l'ordre croissant.



Indice. On peut lire alternativement, parfois dans SectionA, parfois dans SectionB, et on décide à chaque fois quel nom on écrit dans le fichier Promo_MI.

1. Écrire la procédure qui permet de fusionner deux fichiers Fichier1, Fichier2 dans le fichier Fichier3, et s'occupe de l'ouverture et de la fermeture des trois fichiers.
2. Écrire un programme principal pour tester la procédure.

Solution

```

1 PROGRAM FusionFichiers;
2 VAR
3     f1, f2, f3 : text;
4     n1, n2, n3 : string;
5     i : integer;
6
7 Procedure Lire (var f : text; var s : string);
8     Begin
9         if not eof (f) then readln (f, s)
10        else s:= '';
11    End;
12
13 Procedure Fusion (nomf1, nomf2, nomf3 : string);
14     Var f1, f2, f3 : text;
15     m1, m2 : string;
16     Begin
17     assign (f1, nomf1); reset (f1);
18     assign (f2, nomf2); reset (f2);
19     assign (f3, nomf3); rewrite (f3);
20     Lire (f1, m1); Lire (f2, m2);
21     while (m1 <> '') and (m2 <> '') do
22         Begin
23             if m1 < m2 then
24                 begin writeln (f3, m1); Lire (f1, m1); end
25             else
26                 begin writeln (f3, m2); Lire (f2, m2); end;
27             End;
28         while m1 <> '' do
29             begin writeln (f3, m1); Lire(f1, m1); end;
30
31         while m2 <> '' do
32             begin writeln (f3, m2); Lire (f2, m2); end;
33         close (f1);      close (f2);      close (f3);
34     End;
35
36 BEGIN
37 write ('Fichier_1_'); readln (n1);
38 write ('Fichier_2_'); readln (n2);
39 write ('Fichier_3_'); readln (n3);
40 Fusion (n1, n2, n3);
41 END.

```

« Bon courage! »